

REMARKS

The Examiner is thanked for the review of the application. Claims 1-21 are pending. Applicants have taken this opportunity to amend the pending claims to remedy a number of previously undetected informalities. None of the amendments are introduced to overcome the prior art. All amendments are fully supported by the original disclosure. No new matters have been introduced.

Claims Distinguished

Claims 1-2, 4, 9, 11-12, 14, 19 and 21

Claims 1-2, 4, 9, 11-12, 14, 19 and 21 were rejected under 35 USC 102(e) as being anticipated by U.S. Patent No. 6,438,540 to Nasr ("*Nasr*").

35 U.S.C § 102(e) provides:

A person shall be entitled to a patent unless -

...

(e) the invention was described in ... a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent.

To establish a *prima facie* case of anticipation under 35 U.S.C § 102, the Examiner must identify where "each and every facet of the claimed invention is disclosed in the applied reference" *Ex parte Levy*, 17 U.S.P.Q.2d 1461, 1462 (Bd. Pat. App. & Interf. 1990), see also *Electro Med. Sys. S.A. v. Cooper Life Sciences*, 34 F.3d 1048, 1052, 32 U.S.P.Q.2d 1017, 1019 (Fed. Cir. 1994).

Furthermore, anticipation requires that each claim element must be identical to a corresponding element in the applied reference. *Glaverbel Société Anonyme v. Northlake Mktg & Supply, Inc.*, 45 F.3d 1550, 1554 (Fed. Cir. 1995). Indeed, the failure to mention "a claimed element (in) a prior art reference is enough to negate anticipation by that reference" *Atlas Powder Co. v. E.I. duPont De Nemours*, 750 F.2d 1569, 1574 (1984).

Applicants respectfully submit that the Examiner has failed to establish a *prima facie* case of anticipation and that, as amended, Claims 1-2, 4, 9, 11-12, 14, 19 and 21 are patentable over *Nasr*. The Office Action fails to show, and the Applicants are unable to find, where the cited and applied reference discloses, teaches or even suggests the subject matter of the claimed invention.

Among other differences, the cited and applied reference fails to teach, disclose or suggest the element of determining an execution order. In particular, the cited and applied reference fails to teach, disclose or suggest "determining execution order of said actions/computations specified by said data processing cell specifications, from at least one of said data processing cell specifications." The Examiner correctly acknowledges that *Nasr* fails to specify determining an execution order, but proceeds to allege that an execution order is implied by the hierarchical/tree structure taught by *Nasr*. Those of ordinary skill in the art will appreciate that hierarchical tree structures do not necessarily imply execution orders, let alone how to determine them. See for example the attached excerpt from C: How to Program, 2nd Edition, by Deitel et al. (1994) discussing various (inOrder, preOrder and postOrder) tree traversal orders. Accordingly, as will be further discussed below, Claims 1-2, 4, 9, 11-12, 14, 19 and 21 clearly recite one or more of these elements not taught or suggested by *Nasr*.

Nasr teaches a database query system. *Nasr* accomplishes the database queries using a markup language utilizing XML structures. In contrast, Claim 1 reads as follows:

1. (Original) A method of computing comprising:
analyzing a data processing specification having a plurality of data processing cell specifications with each data processing cell specification containing a formula specifying an action or computation;
determining execution order of said actions/computations **specified by said data processing cell specifications**; and
effectuating the data processing specified by the data processing specification **in accordance with the determined execution order** of said actions/computations specified by said data processing cell specifications.
(emphasis added)

The order determination of the method recited in Claim 1 provides a type of order determination not taught or disclosed by *Nasr*. In fact, the Applicants have been unable to identify anywhere within *Nasr*, let alone the cited portions of *Nasr*, where execution order is determined, or implied to be determined. In particular, *Nasr* fails to teach "**determining execution order ... specified by said data processing cell specifications**" (see also page 8, lines 5-9 for further examples). From this distinction alone, it is clear that the present invention, as recited in Claim 1, is distinct from the teachings of *Nasr*. *Nasr* uses standard XML parsing to

determine its handling of queries. *Nasr* is not a system for computing as in the invention of Claims 1. Conversely, the present invention, as recited in Claim 1, is directed to providing such a method of computing with execution orders determined from the data processing cell specifications.

Thus, it is clear that *Nasr* fails to teach the method of computing of the present invention. *Nasr* merely teaches a method of querying a database, using XML queries, that is distinct from the present invention. The query processes of *Nasr* are clearly different from the method of computing of Claim 1 and the corresponding apparatuses of Claims 11 and 21. Therefore, Applicants submit that Claims 1, 11 and 21, as and all the claims dependent therefrom (Claims 2-10 and 12-20) are clearly allowable.

On the other hand, even if the assertion that a hierarchical tree inherently implies an order of execution is correct, which Applicants specific deny, then such a system or method would have no need of an element of determining an execution order, as recited in Claim 1. Accordingly, Applicants submit that Claims 1, 11 and 21, and all the claims dependent therefrom (Claims 2-10 and 12-20) are clearly allowable for this alternate reason as well.

Claims 2-10 and 12-20 depend from allowable independent claims and are therefore allowable for the reasons already noted above. Claims 2-10 and 12-20 are allowable for additional reasons. These claims include further recitations not taught, disclosed, or even suggested by *Nasr*.

For example, Claim 9 recites "specifying one or more global processing characteristics for the specified data processing." Applicants have been unable to identify anywhere in *Nasr* where *Nasr* teaches specifying one or more global processing characteristics for the specified data processing. The cited portion of *Nasr* (Figure 4) has no teaching of global processing characteristics, let alone global processing characteristics for the specified data processing. Accordingly, as *Nasr* fails to teach global processing characteristics for the specified data processing, as in the present invention, it is clear that *Nasr* does not anticipate Claim 9. For these reasons, in addition to those already noted above, Claim 9 and corresponding apparatus Claim 19 are in condition for allowance.

Claims 3, 5, 8, 10, 13, 15, 18, 20 and Claims 6-7 and 16-17

Claims 3, 5-8, 10, 13, 15-18 and 20 were rejected under 35 USC 103(a) as being unpatentable over U.S. Patent No. 6,438,540 to Nasr ("*Nasr*") in view of Published U.S. Patent Application No. 2002/0073080 to Lipkin ("*Lipkin*"). Claims 6-7 and 16-17 were rejected under 35 USC 103(a) as being unpatentable over U.S. Patent No. 6,438,540 to Nasr ("*Nasr*") in view of Published U.S. Patent Application No. 2002/0120685 to Srivastava et al. ("*Srivastava*").

Under § 103, a *prima facie* case of obviousness can be established only if the cited references, alone or in combination, teach each and every element recited in the claim. *In re Bell*, 991 F.2d 781 (Fed. Cir. 1993). Applicants respectfully note that that none of the cited and applied references teach the elements of independent Claims 1, 11 and 21, as already noted above. As each of Claims 3, 5-8, 10, 13, 15-18 and 20 depend from either Claim 1 or Claim 11, they are allowable for this reason alone.

Still further, Applicants submit that the rejections of Claims 3, 5-8, 10, 13, 15-18 and 20 are predicated on combining prior art references that contain no teaching or suggestion of how the cited references could be combined in any manner to form the invention recited in the rejected claims. Simply put, the cited and applied references, taken alone or in combination, do not teach or suggest the subject matter of Claims 3, 5-8, 10, 13, 15-18 and 20.

The Office Action fails to point out any teaching or suggestion in the references related to the **desirability** of the combination suggested in the Office Action. The rejection is using hindsight reasoning based on the **present disclosure** to "produce" the claimed invention by the mere combination of terms similar to those used in Claims 3, 5-8, 10, 13, 15-18 and 20. The references do not teach or suggest how they could be combined in any manner, much less the manner recited in the rejected independent claims. In this regard, the Examiner's attention is directed to the following Federal Circuit and C.C.P.A. decisions:

The teaching or suggestion to make the claimed combination and the reasonable expectation of success must both be found in the prior art, not in applicant's disclosure. *In re Vaeck*, 947 F.2d 488, 20 U.S.P.Q.2d 1438, (Fed. Cir. 1991).

It is wrong to use the patent in suit as a guide through the maze of prior art references, combining the right references in the right way so as to achieve the result of the claims in suit. Monday morning quarterbacking is quite improper when resolving the question of nonobviousness in a court of law. *Orthopedic Equipment, Inc. v. United States*, 217 U.S.P.Q. 193, 199 (Fed. Cir. 1983).

Obviousness cannot be established by combining the teachings of the prior art to produce the claimed invention, absent some teaching or suggestion supporting the combination. Under Section 103, teachings of references can be combined **only** if there is some suggestion or incentive to do so. *ACS Hospital Systems, Inc. v. Montefiore Hospital*, 221 U.S.P.Q. 929, 933 (Fed. Cir. 1984).

The *ACS Hospital Systems, Inc. v. Montefiore Hospital* decision has been cited with approval by the Federal Circuit. See *In re Geiger*, 2 U.S.P.Q. 2d 1276, 1278 (Fed. Cir. 1987). Similar statements have been made in many decisions of the Board of Appeals.

Nor do we see any suggestion in either of the references which would lead anyone having ordinary skill in the art to combine the structure taught by either reference with that taught by the other.

In order to justify a combination of references such as is here suggested it is necessary not only that it be physically possible to combine them, but the art should contain something to suggest the desirability of doing so. Since the art does not suggest the use of either of the patented devices for . . . there is nothing to indicate that one should be modified in view of the other for that purpose. *Ex parte Walker*, 135 U.S.P.Q. 195, 196 (Bd. App. 1962).

We have studied the references and the manner in which the examiner proposes to combine their teachings but we are unable to find in these references any suggestion that they should or could be combined, absent appellant's disclosure in the present application. *Ex parte Lennox*, 144 U.S.P.Q. 224, 225 (Bd. App. 1964).

While as an abstract proposition it might be possible to select features from the secondary references, as the examiner has done, and mechanically combine them with the Mallin device to arrive at appellant's claimed combination, we find absolutely no basis for making such combination neither disclosed nor suggested in the patents relied upon. **In our view only appellant's specification suggests any reasons for combining the features**

of the secondary references with the primary reference and under the provisions of 35 U.S.C. 103 that does not constitute a bar. *Ex parte Fleischmann*, 157 U.S.P.Q. 155 (Bd. App. 1967). (Emphasis added.)

In the instant application, the examiner has done little more than cite references to show that one or more elements or subcombinations thereof, when each is viewed in a vacuum, is known. The claimed invention, however, is clearly directed to a combination of elements. That is to say, appellant does not claim that he has invented one or more new elements but has presented claims to a new combination of elements. **To support the conclusion that the claimed combination is directed to obvious subject matter, either the references must expressly or impliedly suggest the claimed combination or the examiner must present a convincing line of reasoning as to why the artisan would have found the claimed invention to have been obvious in light of the teachings of the references.** *Ex parte Clapp*, 227 U.S.P.Q. 972, 973 (Bd. App. 1985). (Emphasis added.)

In summary, Applicants submit that Claims 3, 5-8, 10, 13, 15-18 and 20 are clearly allowable in view of a lack of teaching or suggestion in the cited and applied references of how they could be combined in any manner, much less in the manner recited in these claims. Furthermore, even if the references were combinable in the manner discussed in the remarks accompanying the rejection of these claims, which Applicants specifically deny, the resultant combination would not meet all of the recitations of the claims, as noted above.

CONCLUSION

As a result, Applicants submit that all of the pending claims are in condition for allowance. Accordingly, a Notice of Allowance is respectfully requested. If the Examiner has any questions concerning the present paper, the Examiner is kindly requested to contact the undersigned at (206) 689-1216. If any fees are due in connection with filing this paper, the Commissioner is authorized to charge Deposit Account No. 500393.

Date: _____

4/16/04

Respectfully submitted,
SCHWABE, WILLIAMSON & WYATT, PC



Adam L.K. Philipp
Reg. No. 42,071

Customer No. 25943
1211 SW Fifth Ave., Ste. 1600-1900
Portland, Oregon 97204
Phone: (206) 622-1711
FAX: (206) 292-0460

E-Mail: aphilipp@schwabe.com

Library of Congress Cataloging-in-Publication Data

Deitel, Harvey M.
C : how to program / H.M. Deitel, P.J. Deitel.. -- 2nd ed.
p. cm.
Includes bibliographical references and index.
ISBN 0-13-226119-7
1. C (Computer program language) I. Deitel, Paul J. II. Title.
QA76.73.C15D44 1994
005.13'3--dc20 93-33697
CIP

Acquisitions Editor: *Marcia Horton*
Production Editor: *Joe Scordato*
Copy Editor: *Michael Schiaparelli*
Chapter Opener and Cover Designer: *Jeannette Jacobs*
Buyer: *Dave Dickey*
Supplements Editor: *Alice Dworkin*
Editorial Assistant: *Dolores Mars*



© 1994, 1992 by Prentice-Hall, Inc.
A Simon & Schuster Company
Englewood Cliffs, New Jersey 07632

The authors and publisher of this book have used their best efforts in preparing this book. These efforts include the development, research, and testing of the theories and programs to determine their effectiveness. The authors and publisher make no warranty of any kind, expressed or implied, with regard to these programs or to the documentation contained in this book. The authors and publisher shall not be liable in any event for incidental or consequential damages in connection with, or arising out of, the furnishing, performance, or use of these programs.

All rights reserved. No part of this book may be reproduced, in any form or by any means, without permission in writing from the publisher.

Printed in the United States of America

10 9 8 7 6

ISBN 0-13-226119-7

Prentice-Hall International (UK) Limited, London
Prentice-Hall of Australia Pty. Limited, Sydney
Prentice-Hall Canada Inc., Toronto
Prentice-Hall Hispanoamericana, S.A., Mexico
Prentice-Hall of India Private Limited, New Delhi
Prentice-Hall of Japan, Inc., Tokyo
Simon & Schuster of Asia Pte. Ltd., Singapore
Editora Prentice-Hall do Brasil, Ltda., Rio de Janeiro

RECEIVED

APR 20 2004

Technology Center 2100

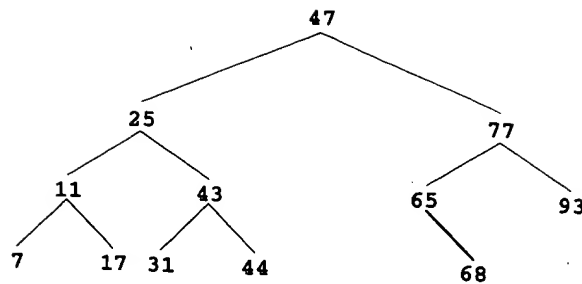


Fig. 12.18 A binary search tree.

The program of Fig. 12.19 (output shown in Fig. 12.20) creates a binary search tree and traverses it three ways—*inorder*, *preorder*, and *postorder*. The program generates 10 random numbers and inserts each in the tree, except that duplicate values are discarded.

The functions used in Fig. 12.19 to create a binary search tree and traverse the tree are recursive. Function `insertNode` receives the address of the tree and an integer to be stored in the tree as arguments. A node can only be inserted as a leaf node in a binary search tree. The steps for inserting a node in a binary search tree are as follows:

- 1) If `*treePtr` is `NULL`, create a new node. Call `malloc`, assign the allocated memory to `*treePtr`, assign to `(*treePtr)->data` the integer to be stored, assign to `(*treePtr)->leftPtr` and `(*treePtr)->rightPtr` the value `NULL`, and return control to the caller (either `main` or a previous call to `insertNode`).
- 2) If the value of `*treePtr` is not `NULL` and the value to be inserted is less than `(*treePtr)->data`, function `insertNode` is called with the address of `(*treePtr)->leftPtr`. Otherwise, function `insertNode` is called with the address of `(*treePtr)->rightPtr`. The recursive steps continue until a `NULL` pointer is found, then step 1) is executed to insert the new node.

Functions `inOrder`, `preOrder`, and `postOrder` each receive a tree (i.e., the pointer to the root node of the tree) and traverse the tree.

The steps for an `inOrder` traversal are:

- 1) Traverse the left subtree `inOrder`.
- 2) Process the value in the node.
- 3) Traverse the right subtree `inOrder`.

The value in a node is not processed until the values in its left subtree are processed. The `inOrder` traversal of the tree in Fig. 12.21 is:

6 13 17 27 33 42 48

Note that the `inOrder` traversal of a binary search tree prints the node values in ascending order. The process of creating a binary search tree actually sorts the data—and thus this process is called the *binary tree sort*.